Symbolic computation of conservation laws for nonlinear partial differential equations in multiple space dimensions

Douglas Poole and Willy Hereman

Department of Mathematical and Computer Sciences, Colorado School of Mines, Golden, CO 80401, USA

Abstract

A method for symbolically computing conservation laws of nonlinear partial differential equations (PDEs) in multiple space dimensions is presented in the language of variational calculus and linear algebra. The steps of the method are illustrated using the Zakharov-Kuznetsov and Kadomtsev-Petviashvili equations as examples.

The method is algorithmic and has been implemented in *Mathematica*. The software package, ConservationLawsMD.m, can be used to symbolically compute and test conservation laws for polynomial PDEs that can be written as nonlinear evolution equations.

The code ConservationLawsMD.m has been applied to multi-dimensional versions of the Sawada-Kotera, Camassa-Holm, Gardner, and Khokhlov-Zabolotskaya equations.

Keywords: Conservation laws; Nonlinear PDEs; Symbolic software; Complete integrability

1. Introduction

Many nonlinear partial differential equations (PDEs) in the applied sciences and engineering are *continuity equations* which express conservation of mass, momentum, energy, or electric charge. Such equations occur in, e.g., fluid mechanics, particle and quantum physics, plasma physics, elasticity, gas dynamics, electromagnetism, magneto-hydro-dynamics, nonlinear optics, etc. Certain nonlinear PDEs admit infinitely many conservation laws. Although most lack a physical interpretation, these conservation laws play an important role in establishing the *complete integrability* of the PDE. Completely integrable PDEs are nonlinear PDEs that can be linearized by some transformation (e.g., the Cole-Hopf transformation linearizes the Burgers equation) or explicitly solved with the Inverse Scattering Transform (IST). See, e.g., Ablowitz and Clarkson (1991).

The search for conservation laws of the Korteweg-de Vries (KdV) equation began around 1964 and the knowledge of conservation laws was paramount for the development of soliton

Email address: `whereman@mines.edu and lpoole@mines.edu` (Douglas Poole and Willy Hereman)

URL: `http://inside.mines.edu/~whereman` (Douglas Poole and Willy Hereman)

theory. As Newell (1983) narrates, the study of conservation laws led to the discovery of the Miura transformation (which connects solutions of the KdV and modified KdV (mKdV) equations) and the Lax pair (Lax, 1968), i.e., a system of linear equations which are only compatible if the original nonlinear PDE holds. In turn, the Lax pair is the starting point for the IST (Ablowitz and Clarkson, 1991; Ablowitz and Segur, 1981) which has been used to construct soliton solutions, i.e., stable solutions that interact elastically upon collision.

Conversely, the existence of many (independent) conserved densities is a *predictor* for complete integrability. The knowledge of conservation laws also aids the study of qualitative properties of PDEs, in particular, bi-Hamiltonian structures and recursion operators (Baldwin and Hereman, 2010). Furthermore, if constitutive properties have been added to close "a model," one should verify that conserved quantities have remained intact. Another application involves numerical solvers for PDEs (Sanz-Serna, 1982), where one checks if the first few (discretized) conserved densities are preserved after each time step.

There are several methods for computing conservation laws as discussed by e.g., Bluman *et al.* (2010), Hereman *et al.* (2005), Naz (2008), Naz *et al.* (2008), and Rosenhaus (2002). One could apply Noether's theorem, which states that a (variational) symmetry of the PDE corresponds to a conservation law. Using Noether's method, the DIFFERENTIAL-GEOMETRY package in *Maple* contains tools for conservation laws developed by Anderson (2004b) and Anderson and Cheb-Terrab (2009). Circumventing Noether's theorem, Wolf (2002) has developed four programs in REDUCE which solve an over-determined system of differential equations to get conservation laws. Based on the integrating factor method, Cheviakov (2007, 2010) has written a *Maple* program that computes a set of integrating factors (multipliers) on the PDE. To find conservation laws, here again, one has to solve a system of differential equations. The *Maple* package PDEtools by Cheb-Terrab and von Bulow (2004) has the commands CONSERVEDCURRENTS and CONSERVEDCURRENTTEST for computing and testing conservation laws using the integrating factor method. Last, conservation laws can be obtained from the Lax operators, as shown by, e.g., Zakharov and Shabat (1972) and Drinfel'd and Sokolov (1985).

By contrast, the method discussed in this paper uses tools from calculus, the calculus of variations, linear algebra, and differential geometry. Briefly, our method works as follows. A candidate (local) density is assumed to be a linear combination with undetermined coefficients of monomials that are invariant under the scaling symmetry of the PDE. Next, the time derivative of the candidate density is computed and evaluated on the PDE. Subsequently, the variational derivative is applied to get a linear system for the undetermined coefficients. The solution of that system is substituted into the candidate density. Once the density is known, the flux is obtained by applying a homotopy operator to invert a divergence. Our method can be implemented in any major computer algebra system (CAS). The package CONSERVATIONLAWSMD.M by Poole and Hereman (2009) is a *Mathematica* implementation based on work by Hereman *et al.* (2005), with new features added by Poole (2009).

This paper is organized as follows. To set the stage, Section 2 shows conservation laws for the Zakharov-Kuznetsov (ZK) and Kadomtsev-Petviashvili (KP) equations. Section 3 covers the tools that will be used in the algorithm. In Section 4, the algorithm is presented

and illustrated for the ZK and KP equations. Section 6 discusses conservation laws of PDEs in multiple space dimensions, including the Khokhlov-Zabolotskaya (KZ) equation and multi-dimensional versions of the Sawada-Kotera (SK), Camassa-Holm (CH) and Gardner equations. Conservation laws for the multi-dimensional SK, CH, and Gardner equations were not found in a literature survey and are presented here for the first time. A general conservation law for the KP equation is given in Section 5. Using the (2+1)-dimensional Gardner equation as an example, Section 7 shows how to use ConservationLawsMD.m. Finally, some conclusions are drawn in Section 8.

2. Examples of Conservation Laws

This paper deals with systems of polynomial PDEs of order M,

$$\boldsymbol{\Delta}(\mathbf{u}^{(M)}(\mathbf{x})) = \mathbf{0}, \tag{1}$$

in n dimensions where $\mathbf{x} = (x^1, x^2, \ldots, x^n)$ is the independent variable. $\mathbf{u}^{(M)}(\mathbf{x})$ denotes the dependent variable $\mathbf{u} = (u^1, \ldots, u^j, \ldots, u^N)$ and its partial derivatives (up to order M) with respect to $\mathbf{x}$. We do not cover systems of PDEs with variable coefficients.

A conservation law for (1) is a scalar PDE in the form

$$\mathrm{Div}\,\mathbf{P} = 0 \quad \text{on } \boldsymbol{\Delta} = \mathbf{0}, \tag{2}$$

where $\mathbf{P} = \mathbf{P}(\mathbf{x}, \mathbf{u}^{(P)}(\mathbf{x}))$ of some order P. The definition follows Olver (1993) and Bluman *et al.* (2010), and is commonly used in literature on symmetries of PDEs. In physics, P is called a conserved current. More precisely, a conservation law can be viewed as an equivalence class of conserved currents (Vinogradov, 1989). Our algorithm computes one member from each equivalence class; usually a representative that is of lowest complexity and free of curl terms.

Since we work on PDEs from the physical sciences, the algorithm and code are restricted to 1 D, 2 D, and 3 D in space, but can be extended to n dimensions. Indeed, many of our applications model dynamical problems, where $\mathbf{x} = (x, y, t)$ for PDEs in 2 D or $\mathbf{x} = (x, y, z, t)$ for PDEs in 3 D in space. In either case, the additional variable, t, denotes time.

Throughout the paper, we will use an alternative definition for (2),

$$\mathcal{D}_t \rho + \mathrm{Div}\,\mathbf{J} = 0 \quad \text{on } \boldsymbol{\Delta} = \mathbf{0}, \tag{3}$$

where $\rho = \rho(\mathbf{x}, \mathbf{u}^{(K)}(\mathbf{x}))$ is the conserved density of some order K, and $\mathbf{J} = \mathbf{J}\left(\mathbf{x}, \mathbf{u}^{(L)}(\mathbf{x})\right)$ is the associated flux of some order L (Miura *et al.*, 1968; Ablowitz and Clarkson, 1991). Comparing (2) with (3), it should be clear that $\mathbf{P} = (\rho, \mathbf{J})$ with $P = \max\{K, L\}$.

For simplicity, in the examples we will denote the dependent variables u^1, u^2, u^3, etc., by u, v, w, etc. Partial derivatives are denoted by subscripts, e.g., $\frac{\partial^{k_1+k_2+k_3} u}{\partial x^{k_1} \partial y^{k_2} \partial z^{k_3}}$ is written as $u_{k_1 x\, k_2 y\, k_3 z}$, where the k_i are non-negative integers. In (3), $\mathrm{Div}\,\mathbf{J}$ is the *total* divergence operator, where $\mathrm{Div}\,\mathbf{J} = \mathcal{D}_x J^x + \mathcal{D}_y J^y$ if $\mathbf{J} = (J^x, J^y)$ and $\mathrm{Div}\,\mathbf{J} = \mathcal{D}_x J^x + \mathcal{D}_y J^y + \mathcal{D}_z J^z$ if $\mathbf{J} = (J^x, J^y, J^z)$. Logically, $\mathcal{D}_t$, $\mathcal{D}_x$, $\mathcal{D}_y$, and $\mathcal{D}_z$ are total derivative operators. For example,

the total derivative operator $\mathcal{D}_x$ (in 1 D) acting on $f = f(x, t, \mathbf{u}^{(M)}(x, t))$ of order M is defined as

$$\mathcal{D}_x f = \frac{\partial f}{\partial x} + \sum_{j=1}^{N} \sum_{k=0}^{M_1^j} u^j_{(k+1)x} \frac{\partial f}{\partial u^j_{kx}}, \tag{4}$$

where M_1^j is the order of f in component u^j and $M = \max\{M_1^1, \ldots, M_1^N\}$. The partial derivative $\frac{\partial}{\partial x}$ acts on any x that appears explicitly in f, but not on u^j or any partial derivatives of u^j. Total derivative operators in multiple dimensions are defined analogously (see Section 3).

The algorithm described in Section 4 allows one to compute local conservation laws for systems of nonlinear PDEs that can be written as evolution equations. For example, if $\mathbf{x} = (x, y, z, t)$, an evolution equation in variable t has the form

$$\mathbf{u}_t = \mathbf{G}(u^1, u^1_x, u^1_y, u^1_z, u^1_{2x}, u^1_{2y}, u^1_{2z}, u^1_{xy}, \ldots, u^N_{M_1^N x\, M_2^N y\, M_3^N z}), \tag{5}$$

where $\mathbf{G}$ is assumed to be smooth and M_1^j, M_2^j, and M_3^j are the orders of component u^j with respect to x, y, and z, respectively, and M is the maximum total order of all terms in the differential function. Few multi-dimensional systems of PDEs are of the form (5). However, it is often possible to obtain a systems of evolution equations by recasting a single higher-order equation into a system of first-order equations, sometimes in conjunction with a simple transformation. If necessary, our program internally interchanges independent variables to obtain (5), where time is the evolution variable. However, that swap of variables is not used in this paper. For a clearer description of the algorithm, we allow systems of evolution equations where *any* component of $\mathbf{x}$ can play the role of evolution variable.

We now introduce two well-documented PDEs together with some of their conservation laws. These PDEs will be used in Section 4 to illustrate the steps of the algorithm.

Example 1. The Zakharov-Kuznetsov (ZK) equation is an evolution equation that models three-dimensional ion-sound solitons in a low pressure uniform magnetized plasma (Zakharov and Kuznetsov, 1974). After re-scaling, it takes the form

$$u_t + \alpha u u_x + \beta(\Delta u)_x = 0, \tag{6}$$

where $u(\mathbf{x}) = u(x, y, z, t), \alpha$ and β are real parameters, and $\Delta = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} + \frac{\partial^2}{\partial z^2}$ is the Laplacian in 3 D. The conservation laws for the (2+1)-dimensional ZK equation,

$$u_t + \alpha u u_x + \beta(u_{2x} + u_{2y})_x = 0, \tag{7}$$

where $u(\mathbf{x}) = u(x, y, t)$, were studied by, e.g., Zakharov and Kuznetsov (1974), Infeld (1985), and Shivamoggi *et al.* (1993). After correcting some of the results reported in Shivamoggi *et al.* (1993), the polynomial conservation laws of (7) are

$$\mathcal{D}_t(u) + \mathcal{D}_x(\tfrac{1}{2}\alpha u^2 + \beta u_{2x}) + \mathcal{D}_y(\beta u_{xy}) = 0, \tag{8}$$

which corresponds to the ZK equation itself, and

$$\mathcal{D}_t\Big(u^2\Big)+\mathcal{D}_x\Big(\tfrac{2}{3}\alpha u^3-\beta(u_x^2-u_y^2)+2\beta u(u_{2x}+u_{2y})\Big)+\mathcal{D}_y\Big(-2\beta u_x u_y\Big)=0, \tag{9}$$

$$\begin{aligned}\mathcal{D}_t\Big(u^3-3\tfrac{\beta}{\alpha}(u_x^2+u_y^2)\Big)&+\mathcal{D}_x\Big(3u^2(\tfrac{1}{4}\alpha u^2+\beta u_{2x})-6\beta u(u_x^2+u_y^2)+3\tfrac{\beta^2}{\alpha}(u_{2x}^2-u_{2y}^2)\\&-6\tfrac{\beta^2}{\alpha}(u_x(u_{3x}+u_{x2y})+u_y(u_{2xy}+u_{3y}))\Big)+\mathcal{D}_y\Big(3\beta u^2u_{xy}+6\tfrac{\beta^2}{\alpha}u_{xy}(u_{2x}+u_{2y})\Big)=0,\end{aligned} \tag{10}$$

$$\begin{aligned}\mathcal{D}_t\Big(tu^2-\tfrac{2}{\alpha}xu\Big)&+\mathcal{D}_x\Big(t(\tfrac{2}{3}\alpha u^3-\beta(u_x^2-u_y^2)+2\beta u(u_{2x}+u_{2y}))-\tfrac{2}{\alpha}x(\tfrac{1}{2}\alpha u^2+\beta u_{2x})+2\tfrac{\beta}{\alpha}u_x\Big)\\&+\mathcal{D}_y\Big(-2\beta(tu_xu_y+\tfrac{1}{\alpha}xu_{xy})\Big)=0.\end{aligned} \tag{11}$$

Note that the fourth conservation law (11) explicitly depends on t and x.

Example 2. The well-known (2+1)-dimensional Kadomtsev-Petviashvili (KP) equation,

$$(u_t+\alpha uu_x+u_{3x})_x+\sigma^2u_{2y}=0, \tag{12}$$

for $u(x,y,t)$, describes shallow water waves with wavelengths much greater than their amplitude moving in the x-direction and subject to weak variations in the y-direction (Kadomtsev and Petviashvili, 1970). The parameter α occurs after a re-scaling of the physical coefficients and $\sigma^2=\pm 1$. Obviously, the KP equation is not an evolution equation. However, it can be written as an evolution system in space variable y,

$$u_y=v, \quad v_y=-\sigma^2(u_{tx}+\alpha u_x^2+\alpha uu_{2x}+u_{4x}). \tag{13}$$

Note that $\frac{1}{\sigma^2}=\sigma^2$, and thus $\sigma^4=1$. System (13) instead of (12) will be used in Section 4. ConservationLawsMD.m has an algorithm that will identify an evolution variable and transform the given PDE into a system of evolution equations.

Equation (12) expresses conservation of momentum:

$$\mathcal{D}_t(u_x)+\mathcal{D}_x(\alpha uu_x+u_{3x})+\mathcal{D}_y(\sigma^2u_y)=0. \tag{14}$$

Other well-documented conservation laws (Wolf, 2002) are

$$\begin{aligned}\mathcal{D}_t\Big(fu\Big)&+\mathcal{D}_x\Big(f(\tfrac{1}{2}\alpha u^2+u_{2x})+(\tfrac{1}{2}\sigma^2f'y^2-fx)(u_t+\alpha uu_x+u_{3x})\Big)\\&+\mathcal{D}_y\Big((\tfrac{1}{2}f'y^2-\sigma^2fx)u_y-f'yu\Big)=0,\end{aligned} \tag{15}$$

$$\begin{aligned}\mathcal{D}_t\Big(fyu\Big)&+\mathcal{D}_x\Big(fy(\tfrac{1}{2}\alpha u^2+u_{2x})+y(\tfrac{1}{6}\sigma^2f'y^2-fx)(u_t+\alpha uu_x+u_{3x})\Big)\\&+\mathcal{D}_y\Big(y(\tfrac{1}{6}f'y^2-\sigma^2fx)u_y-(\tfrac{1}{2}f'y^2-\sigma^2fx)u\Big)=0,\end{aligned} \tag{16}$$

where $f=f(t)$ is an arbitrary function. Thus, there is an infinite family of conservation laws, each of the form (15) or (16). In Section 4 we will show how (8)-(11) are computed straightforwardly with our algorithm. We will also compute several conservation laws for the KP equation. Our current code does not (algorithmically) compute (15) and (16). Instead, conservation laws obtained with the code allow the user to conjecture and test the form of (15) and (16). In Section 5, we give computational details and show how (15) and (16) can be verified.

3. Tools from the Calculus of Variations and Differential Geometry

Three operators from the calculus of variations and differential geometry play a major role in the conservation law algorithm. Namely, the total derivative operator, and the Euler and homotopy operators. All three operators (which act on the jet space) can be defined algorithmically which allows for straightforward and efficient computations.

The algorithm in Section 4 requires that operations applied to *differential functions* $f(\mathbf{x}, \mathbf{u}^{(M)}(\mathbf{x}))$, take place in the jet space, where one component of $\mathbf{x}$ is a parameter.

Although in later sections, one of the space variables will serve as the parameter, in this section we arbitrarily choose t as the parameter (matching (5)). Thus, in all definitions and theorems in this section, 1 D means that there is only one space variable, yet $\mathbf{x} = (x, t)$. Likewise, in 2 D and 3 D cases, $\mathbf{x} = (x, y, t)$ and $\mathbf{x} = (x, y, z, t)$, respectively.

Using (5), we assume that all partial derivatives of $\mathbf{u}$ with respect to t are eliminated from f. Thus, $f(\mathbf{x}, \mathbf{u}^{(M)}(\mathbf{x}))$ with

$$\mathbf{u}^{(M)}(\mathbf{x}) = (u^1, u^1_x, u^1_y, u^1_z, u^1_{2x}, u^1_{2y}, u^1_{2z}, u^1_{xy}, \ldots, u^N_{M_1^N x\, M_2^N y\, M_3^N z}), \tag{17}$$

with M_1^j, M_2^j, M_3^j, and M as defined earlier. Each term in f must be a monomial in jet space variables, either multiplied with a constant or variable coefficient.

Definition 1. The total derivative operator $\mathcal{D}_x$ in 2 D is defined as

$$\mathcal{D}_x f = \frac{\partial f}{\partial x} + \sum_{j=1}^{N} \sum_{k_1=0}^{M_1^j} \sum_{k_2=0}^{M_2^j} u^j_{(k_1+1)x\, k_2 y} \frac{\partial f}{\partial u^j_{k_1 x\, k_2 y}}, \tag{18}$$

where M_1^j and M_2^j are the orders of f for component u^j with respect to x and y, respectively. $\mathcal{D}_y$ is defined analogously. Since t is parameter, $\mathcal{D}_t$ (in 2 D) is defined in a simpler manner,

$$\mathcal{D}_t f = \frac{\partial f}{\partial t} + \sum_{j=1}^{N} \sum_{k_1=0}^{M_1^j} \sum_{k_2=0}^{M_2^j} \frac{\partial f}{\partial u^j_{k_1 x k_2 y}} \mathcal{D}_x^{k_1} \mathcal{D}_y^{k_2} u^j_t. \tag{19}$$

If a total derivative operator were applied by hand to a differential function, $f(\mathbf{x}, \mathbf{u}^{(M)}(\mathbf{x}))$, one would use the product and chain rules to complete the computation. However, formulas like (4), (18), and (19) are more suitable for symbolic computation.

The Euler operator (also known as the variational derivative) plays a fundamental role in the calculus of variations (Olver, 1993), and serves as a key tool in our conservation laws algorithm. The Euler operator can be defined for any number of independent and dependent variables. For example in 1 D, the Euler operator is denoted by $\mathcal{L}_{\mathbf{u}(x)} = (\mathcal{L}_{u^1(x)}, \mathcal{L}_{u^2(x)}, \ldots, \mathcal{L}_{u^j(x)}, \ldots, \mathcal{L}_{u^N(x)})$.

Definition 2. The 1 D Euler operator for dependent variable $u^j(x)$ is defined as

$$\begin{aligned}\mathcal{L}_{u^j(x)} f &= \sum_{k=0}^{M_1^j} (-\mathcal{D}_x)^k \frac{\partial f}{\partial u^j_{kx}} \\ &= \frac{\partial f}{\partial u^j} - \mathcal{D}_x \frac{\partial f}{\partial u^j_x} + \mathcal{D}_x^2 \frac{\partial f}{\partial u^j_{2x}} - \mathcal{D}_x^3 \frac{\partial f}{\partial u^j_{3x}} + \cdots + (-\mathcal{D}_x)^{M_1^j} \frac{\partial f}{\partial u^j_{M_1^j x}},\end{aligned} \tag{20}$$

$j = 1, \ldots, N$. The 2 D and 3 D Euler operators are defined analogously (Olver, 1993). For example, the 2 D Euler operator is

$$\mathcal{L}_{u^j(x,y)} f = \sum_{k_1=0}^{M_1^j} \sum_{k_2=0}^{M_2^j} (-\mathcal{D}_x)^{k_1} (-\mathcal{D}_y)^{k_2} \frac{\partial f}{\partial u_{k_1 x\, k_2 y}}, \quad j = 1, \ldots, N. \tag{21}$$

The Euler operator allows one to test if differential functions are exact which is a key step in the computation of conservation laws.

Definition 3. Let f be a differential function of order M. In 1 D, f is called *exact* if f is a total derivative, i.e., there exists a differential function $F(\mathbf{x}, \mathbf{u}^{(M-1)}(\mathbf{x}))$ such that $f = \mathcal{D}_x F$. In 2 D or 3 D, f is *exact* if f is a total divergence, i.e., there exists a differential vector function $\mathbf{F}(\mathbf{x}, \mathbf{u}^{(M-1)}(\mathbf{x}))$ such that $f = \operatorname{Div} \mathbf{F}$.

Theorem 1. *A differential function f is exact if and only if $\mathcal{L}_{\mathbf{u}(\mathbf{x})} f \equiv \mathbf{0}$. Here, $\mathbf{0}$ is the vector $(0, 0, \cdots, 0)$ which has N components matching the number of components of $\mathbf{u}$.*

Proof. The proof for a general multi-dimensional case is given in, e.g., Poole (2009).

Next, we turn to the homotopy operator (Anderson, 2004a; Olver, 1993), which integrates exact 1 D differential functions, or inverts the total divergence of exact 2 D or 3 D differential functions. Integration routines in CAS have been unreliable when integrating exact differential expressions involving unspecified functions. Often the built-in integration by parts routines fail when arbitrary functions appear in the integrand. The 1 D homotopy operator offers an attractive alternative since it circumvents integration by parts altogether.

Definition 4. Let f be an exact 1 D differential function. The homotopy operator in 1 D is defined (Hereman *et al.*, 2007) as

$$\mathcal{H}_{\mathbf{u}(x)} f = \int_0^1 \left(\sum_{j=1}^{N} \mathcal{I}_{u^j(x)} f \right) [\lambda \mathbf{u}] \frac{d\lambda}{\lambda}, \tag{22}$$

where $\mathbf{u} = (u^1, \ldots, u^j, \ldots, u^N)$. The integrand, $\mathcal{I}_{u^j(x)} f$, is defined as

$$\mathcal{I}_{u^j(x)} f = \sum_{k=1}^{M_1^j} \left(\sum_{i=0}^{k-1} u_{ix}^j \, (-\mathcal{D}_x)^{k-(i+1)} \right) \frac{\partial f}{\partial u_{kx}^j}, \tag{23}$$

where M_1^j is the order of f in dependent variable u^j with respect to x. The notation $f[\lambda \mathbf{u}]$ means that in f one replaces $\mathbf{u}$ by $\lambda\,\mathbf{u}$, $\mathbf{u}_x$ by $\lambda\,\mathbf{u}_x$, and so on for all derivatives of $\mathbf{u}$. λ is an auxiliary parameter that traces the homotopic path.

Given an exact differential function, the 1 D homotopy operator (22) replaces integration by parts (in x) with a sequence of differentiations followed by a standard integration with respect to λ. Indeed, the following theorem states one purpose of the homotopy operator.

Theorem 2. *Let f be exact, i.e., $\mathcal{D}_x F = f$ for some differential function $F(\mathbf{x}, \mathbf{u}^{(M-1)}(\mathbf{x}))$. Then, $F = \mathcal{D}_x^{-1} f = \mathcal{H}_{\mathbf{u}(x)} f$.*

Proof. A proof for the 1 D case in the language of standard calculus is given in Poole and Hereman (2010). See Olver (1993) for a proof based on the variational complex.

The homotopy operator (22) has been a reliable tool for integrating exact *polynomial* differential expressions. For applications, see Cheviakov (2007, 2010); Deconinck and Nivala (2009); Hereman (2006); Hereman *et al.* (2007). However, the homotopy operator fails to integrate certain classes of exact rational expressions as discussed in Poole and Hereman (2010). Although, the homotopy integrator code in Poole and Hereman (2009) covers large classes of exact rational functions, we will not consider rational expressions in this paper.

CAS often cannot invert the divergences of exact 2 D and 3 D differential functions, although some capabilities exist in *Maple*. Again, the homotopy operator is a valuable tool to compute Div^{-1}, when it is impossible to do so by hand or by using the available software tools.

Definition 5. The 2 D homotopy operator is a "vector" operator with two components,

$$\left(\mathcal{H}^{(x)}_{\mathbf{u}(x,y)} f, \mathcal{H}^{(y)}_{\mathbf{u}(x,y)} f \right), \tag{24}$$

where

$$\mathcal{H}^{(x)}_{\mathbf{u}(x,y)} f = \int_0^1 \left(\sum_{j=1}^{N} I^{(x)}_{u^j(x,y)} f \right) [\lambda \mathbf{u}] \, \frac{d\lambda}{\lambda} \quad \text{and} \quad \mathcal{H}^{(y)}_{\mathbf{u}(x,y)} f = \int_0^1 \left(\sum_{j=1}^{N} I^{(y)}_{u^j(x,y)} f \right) [\lambda \mathbf{u}] \, \frac{d\lambda}{\lambda}. \tag{25}$$

The x-integrand, $\mathcal{I}^{(x)}_{u^j(x,y)} f$, is given by

$$I^{(x)}_{u^j(x,y)} f = \sum_{k_1=1}^{M_1^j} \sum_{k_2=0}^{M_2^j} \left(\sum_{i_1=0}^{k_1-1} \sum_{i_2=0}^{k_2} B^{(x)} \, u^j_{i_1 x \, i_2 y} \, (-\mathcal{D}_x)^{k_1-i_1-1} (-\mathcal{D}_y)^{k_2-i_2} \right) \frac{\partial f}{\partial u^j_{k_1 x \, k_2 y}}, \tag{26}$$

with combinatorial coefficient $B^{(x)} = B(i_1, i_2, k_1, k_2)$, where

$$B(i_1, i_2, k_1, k_2) \stackrel{\text{def}}{=} \frac{\binom{i_1+i_2}{i_1}\binom{k_1+k_2-i_1-i_2-1}{k_1-i_1-1}}{\binom{k_1+k_2}{k_1}}. \tag{27}$$

Similarly, the y-integrand, $\mathcal{I}^{(y)}_{u^j(x,y)} f$, is defined as

$$I^{(y)}_{u^j(x,y)} f = \sum_{k_1=0}^{M_1^j} \sum_{k_2=1}^{M_2^j} \left(\sum_{i_1=0}^{k_1} \sum_{i_2=0}^{k_2-1} B^{(y)}\, u^j_{i_1x\, i_2y} \, (-\mathcal{D}_x)^{k_1-i_1} (-\mathcal{D}_y)^{k_2-i_2-1} \right) \frac{\partial f}{\partial u^j_{k_1x\, k_2y}}, \tag{28}$$

where $B^{(y)} = B(i_2, i_1, k_2, k_1)$.

Definition 6. The homotopy operator in 3 D is a three-component vector operator,

$$\left(\mathcal{H}^{(x)}_{\mathbf{u}(x,y,z)} f, \mathcal{H}^{(y)}_{\mathbf{u}(x,y,z)} f, \mathcal{H}^{(z)}_{\mathbf{u}(x,y,z)} f \right), \tag{29}$$

where the x-component is given by

$$\mathcal{H}^{(x)}_{\mathbf{u}(x,y,z)} f = \int_0^1 \left(\sum_{j=1}^{N} \mathcal{I}^{(x)}_{u^j(x,y,z)} f \right) [\lambda \mathbf{u}] \, \frac{d\lambda}{\lambda}. \tag{30}$$

The y- and z-components are defined analogously. The x-integrand is given by

$$\begin{aligned} I^{(x)}_{u^j(x,y,z)} f \;&=\; \sum_{k_1=1}^{M_1^j} \sum_{k_2=0}^{M_2^j} \sum_{k_3=0}^{M_3^j} \sum_{i_1=0}^{k_1-1} \sum_{i_2=0}^{k_2} \sum_{i_3=0}^{k_3} \Big(B^{(x)}\, u^j_{i_1x\, i_2y\, i_3z} \\ &\qquad (-\mathcal{D}_x)^{k_1-i_1-1} (-\mathcal{D}_y)^{k_2-i_2} (-\mathcal{D}_z)^{k_3-i_3} \Big) \frac{\partial f}{\partial u^j_{k_1x\, k_2y\, k_3z}}, \end{aligned} \tag{31}$$

with combinatorial coefficient $B^{(x)} = B(i_1, i_2, i_3, k_1, k_2, k_3)$ where

$$B(i_1, i_2, i_3, k_1, k_2, k_3) \stackrel{\text{def}}{=} \frac{\binom{i_1+i_2+i_3}{i_1}\binom{i_2+i_3}{i_2}\binom{k_1+k_2+k_3-i_1-i_2-i_3-1}{k_1-i_1-1}\binom{k_2+k_3-i_2-i_3}{k_2-i_2}}{\binom{k_1+k_2+k_3}{k_1}\binom{k_2+k_3}{k_2}}. \tag{32}$$

The integrands $I^{(y)}_{u^j(x,y,z)} f$ and $I^{(z)}_{u^j(x,y,z)} f$ are defined analogously. Based on cyclic permutations, they have combinatorial coefficients $B^{(y)} = B(i_2, i_3, i_1, k_2, k_3, k_1)$ and $B^{(z)} = B(i_3, i_1, i_2, k_3, k_1, k_2)$, respectively.

Using homotopy operators, Div^{-1} can be computed based on the following theorem.

Theorem 3. *Let f be exact, i.e., $f = \mathrm{Div}\, \mathbf{F}$ for some $\mathbf{F}(\mathbf{x}, \mathbf{u}^{(M-1)}(\mathbf{x}))$. Then, in the 2 D case, $\mathbf{F} = \mathrm{Div}^{-1} f = \left(\mathcal{H}^{(x)}_{\mathbf{u}(x,y)} f, \mathcal{H}^{(y)}_{\mathbf{u}(x,y)} f \right)$. Analogously, in 3 D one has $\mathbf{F} = \mathrm{Div}^{-1} f = \left(\mathcal{H}^{(x)}_{\mathbf{u}(x,y,z)} f, \mathcal{H}^{(y)}_{\mathbf{u}(x,y,z)} f, \mathcal{H}^{(z)}_{\mathbf{u}(x,y,z)} f \right)$.*

Proof. A proof for the 2 D case is given in Poole (2009). The 3 D case could be proven with similar arguments.

Unfortunately, the outcome of the homotopy operator is not unique. The homotopy integral in the 1 D case has a harmless arbitrary constant. However, in the 2 D and 3 D cases there are infinitely many non-trivial choices for $\mathbf{F}$. From vector calculus we know that $\mathrm{Div}\,\mathrm{Curl}\,\mathbf{K} = 0$. Thus, the addition of $\mathrm{Curl}\,\mathbf{K}$ to $\mathbf{F}$ would not alter $\mathrm{Div}\,\mathbf{F}$. More precisely, for $\mathbf{K} = (\mathcal{D}_y\theta, -\mathcal{D}_x\theta)$ in 2 D, or for $\mathbf{K} = (\mathcal{D}_y\eta - \mathcal{D}_z\xi, \mathcal{D}_z\theta - \mathcal{D}_x\eta, \mathcal{D}_x\xi - \mathcal{D}_y\theta)$ in 3 D, $\mathrm{Div}\,\mathbf{G} = \mathrm{Div}\,(\mathbf{F} + \mathbf{K}) = \mathrm{Div}\,\mathbf{F}$, where θ, η, and ξ are arbitrary functions. To obtain a concise result for Div^{-1}, Poole and Hereman (2010) developed an algorithm that removes curl terms. Furthermore, when f is rational (Poole and Hereman, 2010), the homotopy operator may fail at the singularities of f; but rational functions are not considered in this paper.

4. An Algorithm for Computing a Conservation Law

To compute a conservation law, the PDE is assumed to be in the form given in (5) for a suitable evolution variable. Adhering to (3), if the evolution variable is t, we construct a candidate density. However, if the evolution variable is x, y, or z, we construct a candidate component of the flux corresponding to the evolution variable. For argument's sake let us assume that the evolution variable is time.

The candidate density is constructed by taking a linear combination (with undetermined coefficients) of terms that are invariant under the scaling symmetry of the PDE. The total time derivative of the candidate is computed and evaluated on (5), thus removing all time derivatives from the problem. The resulting expression must be exact, so we use the Euler operator and Theorem 1 to derive the linear system that yields the undetermined coefficients. Substituting these coefficients into the candidate leads to a valid density.

Once the density is known the homotopy operator and Theorems 2 or 3 are used to compute the associated flux, $\mathbf{J}$, taking advantage of (3).

In contrast to other algorithms which attempt to compute the components of $\mathbf{P}$ in (2) all at once, our algorithm computes the density first, followed by the flux. Although restricted to polynomial conservation laws, our constructive method leads to short densities (which are free of divergences and divergence-equivalent terms) and fluxes in which all curl terms are automatically removed.

Definition 7. A term or expression f is a *divergence* if there exists a vector $\mathbf{F}$ such that $f = \mathrm{Div}\,\mathbf{F}$. In the 1 D case, f is a total derivative if there exists a function F such that $f = \mathcal{D}_x F$. Note that $\mathcal{D}_x f$ is essentially a one-dimensional divergence. So, from here onwards, the term "divergence" will also cover the "total derivative" case. Two or more terms are *divergence-equivalent* when a linear combination of the terms is a divergence.

To illustrate the subtleties of the algorithm we intersperse the steps of the algorithm with *two* examples, viz., the ZK and KP equations.

4.1. Computing the Scaling Symmetry

A PDE has a unique set of Lie-point symmetries which may include translations, rotations, dilations, Galilean boosts, and other symmetries (Bluman *et al.*, 2010). The application of such symmetries allows one to generate new solutions from known solutions. We will use only one type of Lie-point symmetry, namely, the scaling or dilation symmetry, to formulate a "candidate density."

Let us assume that a PDE has a scaling symmetry. For example, the ZK equation (7) is invariant under the scaling symmetry

$$(x, y, t, u) \to (\lambda^{-1}x, \lambda^{-1}y, \lambda^{-3}t, \lambda^{2}u), \tag{33}$$

where λ is an arbitrary scaling parameter, not to be confused with λ in Definitions 4 through 6.

Step 1-ZK (Computing the scaling symmetry). To compute (33) with linear algebra, assume that (7) for $u(x, y, t)$ scales uniformly under

$$(x, y, t, u) \to (X, Y, T, U) \equiv (\lambda^{a}x, \lambda^{b}y, \lambda^{c}t, \lambda^{d}u), \tag{34}$$

where $U(X, Y, T)$ and a, b, c, and d are undetermined (rational) exponents. We assume that the parameters α and β do not scale. By the chain rule, (7) transforms into

$$\begin{aligned} &u_t + \alpha u u_x + \beta u_{3x} + \beta u_{x2y} \\ &\quad = \lambda^{c-d}\left(U_T + \alpha\lambda^{a-c-d} U U_X + \beta\lambda^{3a-c} U_{3X} + \beta\lambda^{a+2b-c} U_{X2Y}\right) = 0. \end{aligned} \tag{35}$$

If $a - c - d = 3a - c = a + 2b - c = 0$, we have (7) for $U(X, Y, T)$ up to the scaling factor λ^{c-d}. Setting $a = -1$, we find $b = -1, c = -3$, and $d = 2$, corresponding to (33).

Step 1-KP (Computing the scaling symmetry). The scaling symmetry for the KP equation will be computed similarly. Assume that (13) scales uniformly under

$$(x, y, t, u, v) \to (X, Y, T, U, V) \equiv (\lambda^{a}x, \lambda^{b}y, \lambda^{c}t, \lambda^{d}u, \lambda^{e}v), \tag{36}$$

with unknown rational exponents a through e. Applying the chain rule to get (13) expressed in the variables (X, Y, T, U, V) yields

$$\begin{aligned} &u_y - v = \lambda^{b-d}(U_Y - \lambda^{d-b-e}V) = 0, \\ &v_y + \sigma^2(u_{tx} + u_x^2 + u u_{2x} + u_{4x}) \\ &\quad = \lambda^{b-e}\left(V_Y + \sigma^2\left(\lambda^{a-b+c-d+e}U_{TX} + \alpha\lambda^{2a-b-2d+e}(U_X^2 + U U_{2X}) + \lambda^{4a-b-d+e}U_{4X}\right)\right) = 0. \end{aligned} \tag{37}$$

By setting $d-b-e = a-b+c-d+e = 2a-b-2d+e = 4a-b-d+e = 0$, (37) becomes a scaled version of (13) in the new variables $U(X, Y, T)$ and $V(X, Y, T)$. Setting $a = -1$ yields $b = -2$, $c = -3$, $d = 2$, and $e = 4$. Hence,

$$(x, y, t, u, v) \to (\lambda^{-1}x, \lambda^{-2}y, \lambda^{-3}t, \lambda^{2}u, \lambda^{4}v) \tag{38}$$

is a scaling symmetry of (13).

4.2. Constructing a Candidate Component

Conservation law (2) must hold on solutions of the PDE. Therefore, we search for polynomial conservation laws that obey the scaling symmetry of the PDE. Indeed, we have yet to find a polynomial conservation law that does not adhere to the scaling symmetry.

Based on the scaling symmetry of the PDE, we choose a scaling factor for one of the components of $\mathbf{P}$ in (2). The selected scaling factor will be called the *rank* (R) of that component. Then, we construct a candidate for that component as a linear combination of monomial terms (all of rank R) with undetermined coefficients. By dynamically removing divergence terms and divergence-equivalent terms that candidate is short and of low order.

Step 2-ZK (Building the candidate component). Since the ZK equation (7) has t as evolution variable, we will compute the density ρ of (3) of a fixed rank, for example, $R = 6$.

(a) Construct a list, $\mathcal{P}$, of differential terms containing all powers of dependent variables and products of dependent variables that have rank 6 or less. By (33), u has a scaling factor of 2, so u^3 scales to rank 6 and u^2 has rank 4. This leads to $\mathcal{P} = \{u^3, u^2, u\}$.

(b) Bring all of the terms in $\mathcal{P}$ up to rank 6 and put them into a new list, $\mathcal{Q}$. This is done by applying the total derivative operators with respect to the space variables. Taking the terms in $\mathcal{P}, u^3$ has rank 6 and is placed directly into $\mathcal{Q}$. The term u^2 has rank 4 and can be brought up to rank six in three ways: either by applying $\mathcal{D}_x$ twice, by applying $\mathcal{D}_y$ twice, or by applying each of $\mathcal{D}_x$ and $\mathcal{D}_y$ once, since both $\mathcal{D}_x$ and$\mathcal{D}_y$ have scaling factors of 1. All three possibilities are considered and the resulting terms are put into $\mathcal{Q}$. Similarly, the term u can be brought up to rank 6 in five ways, and all results are placed into $\mathcal{Q}$. Doing so,

$$\mathcal{Q} = \{u^3, u_x^2, uu_{2x}, u_y^2, uu_{2y}, u_x u_y, uu_{xy}, u_{4x}, u_{3xy}, u_{2x2y}, u_{x3y}, u_{4y}\}, \tag{39}$$

in which all monomials are now of rank 6.

(c) With the goal of constructing a nontrivial density with the least number of terms, remove all terms that are *divergences* or are *divergence-equivalent* to other terms in $\mathcal{Q}$. This can be done algorithmically by applying the Euler operator (21) to each term in (39), yielding

$$\mathcal{L}_{u(x,y)}\mathcal{Q} = \{3u^2, -2u_{2x}, 2u_{2x}, -2u_{2y}, 2u_{2y}, -2u_{xy}, 2u_{xy}, 0, 0, 0, 0, 0\}. \tag{40}$$

By Theorem 1, divergences are terms corresponding to 0 in (40). Hence, u_{4x}, u_{3xy}, u_{2x2y}, u_{x3y}, and u_{4y} are divergences and can be removed from $\mathcal{Q}$. Next, all divergence-equivalent terms will be removed. Following Hereman *et al.* (2005), form a linear combination of the terms that remained in (40) with undetermined coefficients p_i, gather like terms, and set it identically equal to zero,

$$3p_1u^2 + 2(p_3 - p_2)u_{2x} + 2(p_5 - p_4)u_{2y} + 2(p_7 - p_6)u_{xy} = 0. \tag{41}$$

Hence, $p_1 = 0$, $p_2 = p_3$, $p_4 = p_5$, and $p_6 = p_7$. Terms with coefficients p_3, p_5, and p_7 are divergence-equivalent to the terms with coefficients p_2. p_4, and p_6, respectively. For each divergence-equivalent pair, the terms of highest order are removed from $\mathcal{Q}$ in (39). After all divergences and divergence-equivalent terms are removed, $\mathcal{Q} = \{u^3, u_x^2, u_y^2, u_x u_y\}$.

(d) A candidate density is obtained by forming a linear combination of the remaining terms in $\mathcal{Q}$ using undetermined coefficients c_i. Thus, the candidate density of rank 6 for (7) is

$$\rho = c_1 u^3 + c_2 u_x^2 + c_3 u_y^2 + c_4 u_x u_y. \tag{42}$$

Now, we turn to the KP equation (12). The conservation laws for the KP equation, (15) and (16), involve an arbitrary functional coefficient $f(t)$. The scaling factor for $f(t)$ depends on the degree if $f(t)$ is polynomial; whereas there is no scaling factor if $f(t)$ is non-polynomial. In general, working with undetermined functional (instead of constant) coefficients $f(x, y, z, t)$ would require a sophisticated solver for PDEs for f (see Wolf (2002)). Therefore, we can *not automatically* compute (15) and (16) with our method. However, our algorithm can find conservation laws with explicit variable coefficients, e.g., tx^2, txy, etc., as long as the degree is specified. Allowing such coefficients causes the candidate component to have a negative rank. By computing several conservation laws with explicit variable coefficients it is possible (by pattern matching) to guess and subsequently test the form of a conservation law with arbitrary functional coefficients.

Step 2-KP (Building a candidate y-component). When the KP equation is replaced by (13), the evolution variable is y. Thus, we will compute a candidate for the y-component of the flux, J^y, in (3). The y-component will have rank equal to -3. The negative rank occurs since differential terms for the component are multiplied by $c_i\, t^m x^n y^p$, which, by (38), scales with $\lambda^{-3m}\lambda^{-n}\lambda^{-2p} = \lambda^{-(3m+n+2p)}$, where m, n, and p are positive integers. The total degree of the variable coefficient $t^m x^n y^p$, is restricted to $0 \le m + n + p \le 3$.

(a) As shown in Table 1, construct two lists, one with all possible coefficients $t^m x^n y^p$ up to degree 3 and the other with differential terms, organized so that the combined rank equals -3. The rank of each term is computed using the scaling factors from (38). For example, t and x have scaling factors of -3 and -1, respectively, so tx^2 has rank -5. Variable u has scaling factor 2, so $t^2 xu$ has rank -3. Since we are computing the y-component of $\mathbf{J}$, the differential terms contain only derivatives with respect to x and t.

Factors of Type $t^m x^n y^p$		Differential Terms		Product
Rank	Coefficient	Rank	Term	Rank
-5	tx^2, xy^2, ty	2	u	-3
-6	y^3, txy, t^2	3	u_x	-3
-7	$t^2 x$, ty^2	4	u^2, u_{2x}, v	-3
-8	$t^2 y$	5	uu_x, u_t, u_{3x}, v_x	-3
-9	t^3	6	u^3, uv, u_x^2, u_{tx}, uu_{2x}, u_{4x}, v_{2x}	-3

Table 1: Factors $t^m x^n y^p$ of degree 3 are paired with differential terms so that their products have ranks -3.

(b) Combine the terms in Table 1 to create a list of all possible terms with rank -3,

$$\mathcal{Q} = \{tx^2u, xy^2u, tyu, y^3u_x, txyu_x, t^2u_x, t^2xu^2, ty^2u^2, t^2xu_{2x}, ty^2u_{2x}, t^2xv, ty^2v \\ t^2yuu_x, t^2yu_t, t^2yu_{3x}, t^2yv_x, t^3u^3, t^3uv, t^3u_x^2, t^3u_{tx}, t^3uu_{2x}, t^3u_{4x}, t^3v_{2x}\}. \tag{43}$$

(c) Remove all divergences and divergence-equivalent terms. Apply the Euler operator to each term in (43). Next, linearly combine the resulting terms to get

$$p_1\begin{pmatrix} tx^2 \\ 0 \end{pmatrix} + p_2\begin{pmatrix} xy^2 \\ 0 \end{pmatrix} + p_3\begin{pmatrix} ty \\ 0 \end{pmatrix} - p_5\begin{pmatrix} ty \\ 0 \end{pmatrix} + p_7\begin{pmatrix} 2t^2xu \\ 0 \end{pmatrix} + p_8\begin{pmatrix} 2ty^2u \\ 0 \end{pmatrix} + p_{11}\begin{pmatrix} 0 \\ t^2x \end{pmatrix} \\ + p_{12}\begin{pmatrix} 0 \\ ty^2 \end{pmatrix} - p_{14}\begin{pmatrix} 2ty \\ 0 \end{pmatrix} + p_{17}\begin{pmatrix} 3t^3u^2 \\ 0 \end{pmatrix} + p_{18}\begin{pmatrix} t^3v \\ t^3u \end{pmatrix} - p_{19}\begin{pmatrix} 2t^3u_{2x} \\ 0 \end{pmatrix} + p_{21}\begin{pmatrix} 2t^3u_{2x} \\ 0 \end{pmatrix} = 0, \tag{44}$$

where the subscript of the undetermined coefficient, p_i, corresponds to the ith term in $\mathcal{Q}$. Missing p_i correspond to terms that are divergences. Gather like terms, set their coefficients equal to zero, and solve the resulting linear system for the p_i, to get $p_1 = p_2 = p_7 = p_8 = p_{11} = p_{12} = p_{17} = p_{18} = 0$, $p_3 = p_5 + 2p_{14}$, and $p_{19} = p_{21}$. Thus, both terms with coefficients p_5 and p_{14} are divergence-equivalent to the term with coefficient p_3. Likewise, the term with coefficient p_{21} is divergence-equivalent to the term with coefficient p_{19}. For each divergence-equivalent pair, the terms with the highest order are removed from (43). After removal of divergences and divergence-equivalent terms

$$\mathcal{Q} = \{tx^2u, xy^2u, tyu, t^2xu^2, ty^2u^2, t^2xv, ty^2v, t^3u^3, t^3uv, t^3u_x^2\}. \tag{45}$$

(d) A linear combination of the terms in (45) with undetermined coefficients c_i yields the candidate (of rank -3) for the y-component of the flux, i.e.,

$$J^y = c_1tx^2u + c_2xy^2u + c_3tyu + c_4t^2xu^2 + c_5ty^2u^2 + c_6t^2xv \\ + c_7ty^2v + c_8t^3u^3 + c_9t^3uv + c_{10}t^3u_x^2. \tag{46}$$

4.3. Evaluating the Undetermined Coefficients

All, part, or none of the candidate density (42) may be an actual density for the ZK equation. It is also possible that the candidate is a linear combination of two or more independent densities, yielding independent conservation laws. The true nature of the density will be revealed by computing the undetermined coefficients. By (3), $\mathcal{D}_t\rho = -\text{Div}\,(J^x, J^y)$, so $\mathcal{D}_t\rho$ must be a divergence with respect to the space variables x and y. Using Theorem 1, an algorithm for computing the undetermined coefficients readily follows.

Step 3-ZK (Computing the undetermined coefficients). To compute the undetermined coefficients, we form a system of linear equations for these coefficients. As part of the solution process, we also generate compatibility conditions for the constant parameters in the PDE, if present.

(a) Compute the total derivative with respect to t of (42),

$$\mathcal{D}_t\rho = 3c_1u^2u_t + 2c_2u_xu_{tx} + 2c_3u_yu_{ty} + c_4(u_{tx}u_y + u_xu_{ty}). \tag{47}$$

Let $E = -\mathcal{D}_t \rho$ after u_t and u_{tx} have been replaced using (7). This yields

$$\begin{aligned} E = {} & 3c_1 u^2(\alpha u u_x + \beta(u_{3x} + u_{x2y})) + 2c_2 u_x(\alpha u u_x + \beta(u_{3x} + u_{x2y}))_x \\ & + 2c_3 u_y(\alpha u u_x + \beta(u_{3x} + u_{x2y}))_y + c_4(u_y(\alpha u u_x + \beta(u_{3x} + u_{x2y}))_x \\ & + u_x(\alpha u u_x + \beta(u_{3x} + u_{x2y}))_y). \end{aligned} \tag{48}$$

(b) By (3), $E = \mathrm{Div}\,(J^x, J^y)$. Therefore, by Theorem 1, $\mathcal{L}_{u(x,y)}\ E \equiv \mathbf{0}$. Apply the Euler operator to (48), gather like terms, and set the result identically equal to zero:

$$\begin{aligned} 0 \equiv \mathcal{L}_{u(x,y)}\ E \ = \ & -2\big((3c_1\beta + c_3\alpha)u_x u_{2y} + 2(3c_1\beta + c_3\alpha)u_y u_{xy} \\ & + 2c_4\alpha u_x u_{xy} + c_4\alpha u_y u_{2x} + 3(3c_1\beta + c_2\alpha)u_x u_{2x}\big). \end{aligned} \tag{49}$$

(c) Form a *linear* system of equations for the undetermined coefficients c_i by setting each coefficient equal to zero, thus satisfying (49). After eliminating duplicate equations, the system is

$$3c_1\beta + c_3\alpha = 0, \qquad c_4\alpha = 0, \qquad 3c_1\beta + c_2\alpha = 0. \tag{50}$$

(d) Check for possible compatibility conditions on the parameters α and β in (50). This is done by setting each $c_i = 1$, one at a time, and algebraically eliminating the other undetermined coefficients. Consult Göktaş and Hereman (1997) for details about searching for compatibility conditions. System (50) is compatible for all nonzero α and β.

(e) Solve (50), taking into account the compatibility conditions (if applicable). Here,

$$c_2 = c_3 = -3\tfrac{\beta}{\alpha} c_1, \qquad c_4 = 0, \tag{51}$$

where c_1 is arbitrary. We set $c_1 = 1$ so that the density is normalized on the highest degree term, yielding

$$\rho = u^3 - 3\tfrac{\beta}{\alpha}(u_x^2 + u_y^2). \tag{52}$$

Step 3-KP (Computing the undetermined coefficients). The procedure to find the undetermined coefficients in the KP case is similar to that of the ZK case.

(a) Starting from (46), compute

$$\begin{aligned} \mathcal{D}_y J^y = {} & (c_1 x + 2c_4 tu)txu_y + c_2 xy(2u + yu_y) + (c_3 t + 2c_5 tyu)(u + yu_y) \\ & + c_6 t^2 x v_y + c_7 ty(2v + yv_y) + 3c_8 t^3 u^2 u_y + c_9 t^3(u_y v + u v_y) + 2c_{10} t^3 u_x u_{xy}, \end{aligned} \tag{53}$$

and replace u_y and v_y and their differential consequences using (13). Thus,

$$\begin{aligned} E = -\mathcal{D}_y J^y = {} & -(c_1 x + 2c_4 tu)txv - c_2 xy(2u + yv) - (c_3 t + 2c_5 tyu)(u + yv) \\ & + \sigma^2(c_6 t^2 x + c_7 ty^2 + c_9 t^3 u)(u_{tx} + \alpha u_x^2 + \alpha u u_{2x} + u_{4x}) - 2c_7 tyv \\ & - 3c_8 t^3 u^2 v - c_9 t^3 v^2 - 2c_{10} t^3 u_x v_x. \end{aligned} \tag{54}$$

(b) Apply the Euler operator to (54) and set the result identically equal to zero. This yields

$$
\begin{aligned}
(0,\,0) &= \mathbf{0} \equiv \mathcal{L}_{\mathbf{u}(t,x)}\ E = \Big(\mathcal{L}_{u(t,x)}\ E,\ \mathcal{L}_{v(t,x)}\ E\Big) \\
&= -\Big(2c_2xy + (c_3 - 2\sigma^2 c_6)t + 2c_4t^2xv + 2c_5ty(2u + yv) + 6c_8t^3uv \\
&\quad - 2\sigma^2 c_9 t^2(\tfrac{3}{2}u_x + tu_{tx} + \alpha t u_x^2 + \alpha u u_{2x} + tu_{4x}) - 2c_{10}t^3v_{2x},\ c_1tx^2 + c_2xy^2 \\
&\quad + (c_3 + 2c_7)ty + 2c_4t^2xu + 2c_5ty^2u + 3c_8t^3u^2 + 2c_9t^3v - 2c_{10}t^3u_{2x}\Big). \qquad (55)
\end{aligned}
$$

(c) Form a linear system for the undetermined coefficients c_i. After duplicate equations and common factors have been removed, one gets

$$
c_1 = 0,\ c_2 = 0,\ c_3 - 2\sigma^2 c_6 = 0,\ c_3 + 2c_7 = 0,\ c_4 = 0,\ c_5 = 0,\ c_8 = 0,\ c_9 = 0,\ c_{10} = 0. \qquad (56)
$$

(d) Compute potential compatibility conditions on the parameters α and σ. Again, the system is compatible for all nonzero values of α and σ.

(e) Use $\sigma^2 = \pm 1$ and solve the linear system, yielding

$$
c_1 = c_2 = c_4 = c_5 = c_8 = c_9 = c_{10} = 0, \quad c_6 = \tfrac{1}{2}\sigma^2 c_3, \quad c_7 = -\tfrac{1}{2}c_3. \qquad (57)
$$

Set $c_3 = -2$ (to normalize the density) and substitute the result into (46), to obtain

$$
J^y = -t(2yu + (\sigma^2 tx - y^2)v), \qquad (58)
$$

which matches J^y in (15) if $f(t) = t^2$ and $v = u_y$.

4.4. Completing the Conservation Law

With the density (or a component of the flux at hand), the remaining components of the conservation law can be computed with the homotopy operator using Theorem 2 or 3.

Step 4-ZK (Computing the flux, $\mathbf{J}$). Again, by the continuity equation (3), $\mathrm{Div}\,\mathbf{J} = \mathrm{Div}\,(J^x, J^y) = -\mathcal{D}_t\rho = E$. Therefore, compute $\mathrm{Div}^{-1}\,E$, where the divergence is with respect to x and y. After substitution of (51) with $c_1 = 1$ into (48),

$$
\begin{aligned}
E &= 3u^2(\alpha u u_x + \beta u_{3x} + \beta u_{x2y}) - 6\tfrac{\beta}{\alpha}u_x(\alpha u u_x + \beta u_{3x} + \beta u_{x2y})_x \\
&\quad - 6\tfrac{\beta}{\alpha}u_y(\alpha u u_x + \beta u_{3x} + \beta u_{x2y})_y. \qquad (59)
\end{aligned}
$$

Apply the 2 D homotopy operator from Theorem 3. Compute the integrands (26) and (28):

$$
\begin{aligned}
I^{(x)}_{u(x,y)}E &= 3\alpha u^4 + \beta\left(9u^2(u_{2x} + \tfrac{2}{3}u_{2y}) - 6u(3u_x^2 + u_y^2)\right) + \tfrac{\beta^2}{\alpha}\left(6u_{2x}^2 + 5u_{xy}^2 + \tfrac{3}{2}u_{2y}^2\right. \\
&\quad \left.+\tfrac{3}{2}u(u_{2x2y} + u_{4y}) - u_x(12u_{3x} + 7u_{x2y}) - u_y(3u_{3y} + 8u_{2xy}) + \tfrac{5}{2}u_{2x}u_{2y}\right), \quad (60) \\
I^{(y)}_{u(x,y)}E &= 3\beta u(uu_{xy} - 4u_xu_y) - \tfrac{1}{2}\tfrac{\beta^2}{\alpha}\left(3u(u_{3xy} + u_{x3y}) + u_x(13u_{2xy} + 3u_{3y})\right. \\
&\quad \left.+ 5u_y(u_{3x} + 3u_{x2y}) - 9u_{xy}(u_{2x} + u_{2y})\right), \qquad (61)
\end{aligned}
$$

respectively. Use (25), to compute $\hat{\mathbf{J}} = \left(\mathcal{H}^{(x)}_{\mathbf{u}(x,y)} E, \mathcal{H}^{(y)}_{\mathbf{u}(x,y)} E\right)$ where

$$
\begin{aligned}
\mathcal{H}^{(x)}_{\mathbf{u}(x,y)} E &= \int_0^1 \left(\mathcal{I}^{(x)}_{u(x,y)} E\right)[\lambda \mathbf{u}] \frac{d\lambda}{\lambda} \\
&= \tfrac{3}{4}\alpha u^4 + \beta\left(3u^2(u_{2x} + \tfrac{2}{3}u_{2y}) - 2u(3u_x^2 + u_y^2)\right) + \tfrac{\beta^2}{\alpha}\left(3u_{2x}^2 + \tfrac{5}{2}u_{xy}^2 + \tfrac{3}{4}u_{2y}^2\right. \\
&\quad \left. + \tfrac{3}{4}u(u_{2x2y} + u_{4y}) - u_x(6u_{3x} + \tfrac{7}{2}u_{x2y}) - u_y(\tfrac{3}{2}u_{3y} + 4u_{2xy}) + \tfrac{5}{4}u_{2x}u_{2y}\right), \quad (62) \\
\mathcal{H}^{(y)}_{\mathbf{u}(x,y)} E &= \int_0^1 \left(\mathcal{I}^{(x)}_{u(x,y)} E\right)[\lambda \mathbf{u}] \frac{d\lambda}{\lambda} \\
&= \beta u(u u_{xy} - 4u_x u_y) - \tfrac{1}{4}\tfrac{\beta^2}{\alpha}\left(3u(u_{3xy} + u_{x3y}) + u_x(13u_{2xy} + 3u_{3y})\right. \\
&\quad \left. + 5u_y(u_{3x} + 3u_{x2y}) - 9u_{xy}(u_{2x} + u_{2y})\right). \quad (63)
\end{aligned}
$$

Notice that $\hat{\mathbf{J}}$ has a curl term, $\mathbf{K} = (\mathcal{D}_y \theta, -\mathcal{D}_x \theta)$, with

$$
\theta = 2\beta u^2 u_y + \tfrac{1}{4}\tfrac{\beta^2}{\alpha}\Big(3u(u_{2xy} + u_{3y}) + 5(2u_x u_{xy} + 3u_y u_{2y} + u_{2x}u_y)\Big). \quad (64)
$$

Therefore, compute $\hat{\mathbf{J}} - \mathbf{K}$ to obtain

$$
\begin{aligned}
J^x &= 3\Big(u^2(\tfrac{1}{4}\alpha u^2 + \beta u_{2x}) - 2\beta u(u_x^2 + u_y^2) + \tfrac{\beta^2}{\alpha}(u_{2x}^2 - u_{2y}^2) \\
&\qquad - 2\tfrac{\beta^2}{\alpha}(u_x(u_{3x} + u_{x2y}) + u_y(u_{2xy} + u_{3y}))\Big), \quad (65) \\
J^y &= 3\beta\Big(u^2 u_{xy} + 2\tfrac{\beta}{\alpha}u_{xy}(u_{2x} + u_{2y})\Big), \quad (66)
\end{aligned}
$$

which match the components in (10).

Step 4-KP (Computing the density and the x-component of the flux). For the KP example, (ρ, J^x) remains to be computed. Using the continuity equation (3), $\mathcal{D}_t\rho + \mathcal{D}_x J^x = -\mathcal{D}_y J^y = E$. Thus, to find (ρ, J^x), compute $\mathrm{Div}^{-1} E$, where this time the divergence is with respect to t and x. Proceed as in the previous example. First, substitute (57) and $c_3 = -2$ into (54),

$$
E = t\Big(2u + \left(\sigma^2 y^2 - tx\right)\left(u_{tx} + \alpha u_x^2 + \alpha u u_{2x} + u_{4x}\right)\Big). \quad (67)
$$

Second, compute the integrands for the homotopy operator,

$$
\begin{aligned}
I^{(t)}_{u(t,x)} E &= -\tfrac{1}{2}(u\mathcal{D}_x - u_x\mathcal{I})\frac{\partial E}{\partial u_{tx}} = \tfrac{1}{2}t(tu + (\sigma^2 y^2 - tx)u_x), \quad (68) \\
I^{(t)}_{v(t,x)} E &= 0, \quad (69) \\
I^{(x)}_{u(t,x)} E &= u\frac{\partial E}{\partial u_x} - \tfrac{1}{2}(u\mathcal{D}_t - u_t\mathcal{I})\frac{\partial E}{\partial u_{tx}} - (u\mathcal{D}_x - u_x\mathcal{I})\frac{\partial E}{\partial u_{2x}} - (u\mathcal{D}_x^3 - u_x\mathcal{D}_x^2 + u_{2x}\mathcal{D}_x - u_{3x}\mathcal{I})\frac{\partial E}{\partial u_{4x}} \\
&= t^2(\alpha u^2 + u_{2x}) + t(\sigma^2 y^2 - tx)(\tfrac{1}{2}u_t + 2\alpha u u_x + u_{3x}) - (\tfrac{1}{2}\sigma^2 y^2 - tx)u, \quad (70) \\
I^{(x)}_{v(t,x)} E &= 0. \quad (71)
\end{aligned}
$$

Next, compute

$$
\begin{aligned}
\hat{\rho} &= \mathcal{H}^{(t)}_{\mathbf{u}(t,x)} E = \int_0^1 \left(I^{(t)}_{u(t,x)} E + I^{(t)}_{v(t,x)} E \right) [\lambda u] \frac{d\lambda}{\lambda} \\
&= \int_0^1 \left(\tfrac{1}{2} t (tu + (\sigma^2 y^2 - tx) u_x) \right) d\lambda = \tfrac{1}{2} t (tu + (\sigma^2 y^2 - tx) u_x), \qquad (72) \\
\hat{J}^x &= \mathcal{H}^{(x)}_{\mathbf{u}(t,x)} E = \int_0^1 \left(I^{(x)}_{u(t,x)} E + I^{(x)}_{v(t,x)} E \right) [\lambda u] \frac{d\lambda}{\lambda} \\
&= \int_0^1 \left(t^2 (\alpha \lambda u^2 + u_{2x}) + t(\sigma^2 y^2 - tx)(\tfrac{1}{2} u_t + 2\alpha \lambda u u_x + u_{3x}) - (\tfrac{1}{2}\sigma^2 y^2 - tx) u \right) d\lambda \\
&= t^2 (\tfrac{1}{2} \alpha u^2 + u_{2x}) + t(\sigma^2 y^2 - tx)(\tfrac{1}{2} u_t + \alpha u u_x + u_{3x}) - (\tfrac{1}{2} \sigma^2 y^2 - tx) u, \qquad (73)
\end{aligned}
$$

and remove the curl term $\mathbf{K} = (\mathcal{D}_x \theta, -\mathcal{D}_t \theta)$ with $\theta = \frac{1}{2} t (\sigma^2 y^2 - tx) u$, to obtain

$$
\rho = t^2 u, \quad J^x = t^2 (\tfrac{1}{2} \alpha u^2 + u_{2x}) + t(\sigma^2 y^2 - tx)(u_t + \alpha u u_x + u_{3x}). \qquad (74)
$$

The computed conservation law is the same as (15) where $f(t) = t^2$ and $v = u_y$.

5. A Generalized Conservation Law for the KP Equation

Due to the presence of an arbitrary function $f(t)$, it is impossible to *algorithmically* compute (15) with our code. The generalization of (74) to (15) is based on inspection of the conservation laws in Table 2 as computed by our program ConservationLawsMD.m. Indeed, pattern matching with the results in Table 2 and some interactive work lead to (15),

Rank	Conservation Law
5	$\mathcal{D}_t \Big(u \Big) + \mathcal{D}_x \Big(\frac{1}{2} \alpha u^2 + u_{2x} - x(u_t + \alpha u u_x + u_{3x}) \Big) + \mathcal{D}_y \Big(- \sigma^2 x v \Big) = 0$
2	$\mathcal{D}_t \Big(tu \Big) + \mathcal{D}_x \Big(t(\frac{1}{2} \alpha u^2 + u_{2x}) + (\frac{1}{2} \sigma^2 y^2 - tx)(u_t + \alpha u u_x + u_{3x}) \Big)$ $+ \mathcal{D}_y \Big((\frac{1}{2} y^2 - \sigma^2 t x) v - yu \Big) = 0.$
-1	$\mathcal{D}_t \Big(t^2 u \Big) + \mathcal{D}_x \Big(t^2 (\frac{1}{2} \alpha u^2 + u_{2x}) + t(\sigma^2 y^2 - tx)(u_t + \alpha u u_x + u_{3x}) \Big)$ $+ \mathcal{D}_y \Big(t(y^2 - \sigma^2 t x) v - 2tyu \Big) = 0$
-4	$\mathcal{D}_t \Big(t^3 u \Big) + \mathcal{D}_x \Big(t^3 (\frac{1}{2} \alpha u^2 + u_{2x}) + t^2 (\frac{3}{2} \sigma^2 y^2 - tx)(u_t + \alpha u u_x + u_{3x}) \Big)$ $+ \mathcal{D}_y \Big(t^2 \Big((\frac{3}{2} y^2 - \sigma^2 t x) v - 3yu \Big) \Big) = 0$

Table 2: Additional conservation laws for the KP equation (13).

which can be then be verified with ConservationLawsMD.m as follows.

The conservation laws in Table 2 suggest that a density has the form $t^n u$, or more general, $f(t) u$, where $f(t)$ is an arbitrary function. The corresponding flux would be harder to guess.

However, it can be computed as follows. Since the KP equation (13) is an evolution equation in y, we construct a suitable candidate for J^y. Guided by the results in Table 2, we take

$$J^y = c_1 f'(t) y u + c_2 f'(t) y^2 v + c_3 f(t) x v, \tag{75}$$

where c_1, c_2, and c_3 are undetermined coefficients, and u_y is replaced by v in agreement with (13). As before, we compute $\mathcal{D}_y J^y$ and replace u_y and v_y using (13). Doing so,

$$E = \mathcal{D}_y P^y = c_1 f' u + (c_1 + 2c_2) f' y v - (\sigma^2 c_2 f' y^2 + \sigma^2 c_3 f x)(u_{tx} + \alpha u_x^2 + \alpha u u_{2x} + u_{4x}). \tag{76}$$

By (3), $\mathcal{D}_y J^y = -\mathrm{Div}(\rho, J^x)$. By Theorem 1,

$$(0,\,0) = \mathbf{0} \equiv \mathcal{L}_{\mathbf{u}(t,x)} E = \big((c_1 - \sigma^2 c_3) f',\, (c_1 + 2c_2) f' y\big)\,. \tag{77}$$

Clearly, $c_2 = -\frac{1}{2} c_1$ and $c_3 = \sigma^2 c_1$. If we set $c_1 = -1$ and $v = u_y$ we obtain J^y in (15). Application of the homotopy operator (in this case to an expression with arbitrary functional coefficients) yields (ρ, J^x). This is how conservation law (15) was computed. Conservation law (16) was obtained in a similar way. Both conservation laws were then verified using the ConservationLawsMD.m code.

6. Applications

In this section we state results obtained by using our algorithm on a variety of (2+1)- and (3+1)-dimensional nonlinear PDEs. The selected PDEs highlight several of the issues that arise when using our algorithm and software package ConservationLawsMD.m.

6.1. The Sawada-Kotera Equation in 2 D

The (2+1)-dimensional SK equation (Konopelchenko and Dubrovsky, 1984),

$$u_t = 5u^2 u_x + 5u u_{3x} + 5u u_y + 5u_x u_{2x} + 5u_{2xy} + u_{5x} - 5\partial_x^{-1} u_{2y} + 5u_x \partial_x^{-1} u_y, \tag{78}$$

with $u(\mathbf{x}) = u(x, y, t)$ is a *completely integrable* 2 D generalization of the standard SK equation. The latter has infinitely many conservation laws (see, e.g., Göktaş and Hereman (1997)). Our algorithm can not handle the integral terms in (78), so we set $v = \partial_x^{-1} u_y$. Doing so, (78) becomes a system of evolution equations in y:

$$v_y = -\tfrac{1}{5} u_t + u^2 u_x + u u_{3x} + u v_x + u_x u_{2x} + v_{3x} + \tfrac{1}{5} u_{5x} + u_x v, \quad u_y = v_x. \tag{79}$$

Application of our algorithm to (79) yields several conservation laws, all of which have densities u, tu, $t^2 u$, etc., and yu, tyu, $t^2 yu$, etc. Like with the KP equation, this suggests that there are conservation laws with an arbitrary functional coefficient $f(t)$. Proceeding as in Section 5 and using ConservationLawsMD.m, we obtained

$$\mathcal{D}_t\Big(fu\Big) + \mathcal{D}_x\Big(f' y v - 5f(\tfrac{1}{3}u^3 + uv + u u_{2x} + u_{xy} + \tfrac{1}{5}u_{4x})\Big) + \mathcal{D}_y\Big(5fv - f' y u\Big) = 0, \tag{80}$$

$$\mathcal{D}_t\Big(fyu\Big) + \mathcal{D}_x\Big((\tfrac{1}{2} f' y^2 - 5fx) v - 5fy(\tfrac{1}{3}u^3 + uv + u u_{2x} + u_{xy} + \tfrac{1}{5}u_{4x})\Big)$$
$$+ \mathcal{D}_y\Big(5fyv - (\tfrac{1}{2} f' y^2 - 5fx) u\Big) = 0. \tag{81}$$

Note that the densities in (80) and (81) are identical to those in (15) and (16) for the KP equation. These two densities occur often in (2+1)-dimensional PDEs that have a u_{tx} instead of a u_t term, as shown in the next example.

6.2. The Khokhlov-Zabolotskaya Equation in 2D and 3D

The Khokhlov-Zabolotskaya (KZ) equation or dispersionless KP equation describes the propagation of sound in non-linear media in two or three space dimensions (Sanders and Wang, 1997a). The (2+1)-dimensional KZ equation,

$$(u_t - uu_x)_x - u_{2y} = 0, \tag{82}$$

with $u(\mathbf{x}) = u(x, y, t)$ can be written as a system of evolution equations in y,

$$u_y = v, \quad v_y = u_{tx} - u_x^2 - uu_{2x}, \tag{83}$$

by setting $v = u_y$. Again, two familiar densities appear in the following conservation laws, computed indirectly as we showed for the KP and SK equations,

$$\mathcal{D}_t(u_x) + \mathcal{D}_x(-uu_x) + \mathcal{D}_y(-u_y) = 0, \tag{84}$$

$$\mathcal{D}_t\Big(fu\Big) + \mathcal{D}_x\Big(-\tfrac{1}{2}fu^2 - (\tfrac{1}{2}f'y^2 + fx)(u_t - uu_x)\Big) + \mathcal{D}_y\Big((\tfrac{1}{2}f'y^2 + fx)u_y - f'yu\Big) = 0, \tag{85}$$

$$\begin{aligned} &\mathcal{D}_t\Big(fyu\Big) + \mathcal{D}_x\Big(-\tfrac{1}{2}fyu^2 - y(\tfrac{1}{6}f'y^2 + fx)(u_t - uu_x)\Big) \\ &\quad + \mathcal{D}_y\Big(y(\tfrac{1}{6}f'y^2 + fx)u_y - (\tfrac{1}{2}f'y^2 + fx)u\Big) = 0, \end{aligned} \tag{86}$$

where $f(t)$ is an arbitrary function. Actually, (85) and (86) are *nonlocal* because, from (82), $u_t - uu_x = \int u_{2y}\, dx$. By swapping terms in the density and the x-component of the flux, (85) with $f(t) = 1$, can be rewritten as

$$\mathcal{D}_t\Big(xu_x\Big) + \mathcal{D}_x\Big(\tfrac{1}{2}u^2 - xuu_x\Big) + \mathcal{D}_y\Big(-xu_y\Big) = 0, \tag{87}$$

which is local. The computation of conservation laws for the (3+1)-dimensional KZ equation,

$$(u_t - uu_x)_x - u_{2y} - u_{2z} = 0, \tag{88}$$

where $u(\mathbf{x}) = u(x, y, z, t)$, is more difficult. This equation can be written as a system of evolution equations in either y or z. Although the intermediate results differ, either choice leads to equivalent conservation laws. Writing (88) as an evolution system in z,

$$u_z = v, \quad v_z = u_{tx} - u_x^2 - uu_{2x} - u_{2y}, \tag{89}$$

ConservationLawsMD.m is able to compute a variety of conservation laws whose densities are shown in Table 3.

Rank	Densities Explicitly Dependent on x, y, z
2	$\rho_1 = xu_x$
0	$\rho_2 = xyu_x, \quad \rho_3 = xzu_x$
-1	$\rho_4 = tu$
-2	$\rho_5 = xyzu_x, \quad \rho_6 = x(y^2 - z^2)u_x$
-3	$\rho_7 = tyu, \quad \rho_8 = tzu$
-4	$\rho_9 = t^2u, \quad \rho_{10} = xy(y^2 - 3z^2)u_x, \quad \rho_{11} = xz^2(3y - z)u_x$
-5	$\rho_{12} = tyzu, \quad \rho_{13} = t(y^2 - z^2)u$
-6	$\rho_{14} = t^2yu, \quad \rho_{15} = t^2xzu, \quad \rho_{16} = xyz(y^2 - z^2)u_x, \quad \rho_{17} = x(y^4 - 6y^2z^2 + z^4)u_x$
-7	$\rho_{18} = t^3u, \quad \rho_{19} = ty(y^2 - 3z^2)u, \quad \rho_{20} = tz(3y^2 - z^2)u$
-8	$\rho_{21} = t^2yzu, \quad \rho_{22} = t^2(y^2 - z^2)u, \quad \rho_{23} = xy(y^4 - 10y^2z^2 + 5z^4)u_x,$ $\rho_{24} = xz(5y^4 - 10y^2z^2 + z^4)u_x$

Table 3: Densities for the (3+1)-dimensional KZ equation (88).

Density $\rho_1 = xu_x$ in Table 3 is part of local conservation law

$$\mathcal{D}_t\Big(xu_x\Big) + \mathcal{D}_x\Big(\tfrac{1}{2}u^2 - xuu_x\Big) + \mathcal{D}_y\Big(-xu_y\Big) + \mathcal{D}_z\Big(-xu_z\Big) = 0, \tag{90}$$

which can be rewritten as a nonlocal conservation law

$$\mathcal{D}_t\Big(u\Big) + \mathcal{D}_x\Big(-\tfrac{1}{2}u^2 - x(u_t - uu_x)\Big) + \mathcal{D}_y\Big(xu_y\Big) + \mathcal{D}_z\Big(xu_z\Big) = 0. \tag{91}$$

In general, if a factor xu_x appears in a density then that factor can be replaced by u. Doing so, all densities in Table 3 that can be expressed as $\rho = g(y, z, t)u$, where $g(y, z, t)$ is arbitrary. Introducing an arbitrary function $h = h(y, z, t)$, the conservation laws corresponding to the densities in Table 3 can be summarized as

$$\begin{aligned}&\mathcal{D}_t\Big(gu\Big) + \mathcal{D}_x\Big(-\tfrac{1}{2}gu^2 - (xg + h)(u_t - uu_x)\Big) + \mathcal{D}_y\Big((xg + h)u_y - (xg_y + h_y)u\Big)\\ &+\mathcal{D}_z\Big((xg + h)u_z - (xg_z + h_z)u\Big) = -\Big(h_{2y} + h_{2z} - g_t + x(g_{2y} + g_{2z})\Big)u.\end{aligned} \tag{92}$$

Equation (92) is only a conservation law when the constraints $\Delta g = 0$ and $\Delta h = g_t$ are satisfied, where $\Delta = \frac{\partial^2}{\partial y^2} + \frac{\partial^2}{\partial z^2}$. Thus, g must be a harmonic function and h must satisfy the Poisson equation with g_t on the right hand side. Combining both equations produces the biharmonic equation $\Delta^2 h = 0$. As shown by Tikhonov and Samarskii (1963), $\Delta^2 h = 0$ has general solutions of the form

$$h = y\, h_1(y, z) + h_2(y, z) \quad \text{and} \quad h = z\, h_1(y, z) + h_2(y, z), \tag{93}$$

where $\Delta h_1 = 0$ and $\Delta h_2 = 0$. Treating t as a parameter, four solutions for $h(y, z, t)$ are

$$h(y, z, t) = \tfrac{1}{2}y\, \partial_y^{-1} g_t(y, z, t), \tag{94}$$

$$h(y, z, t) = \tfrac{1}{2}\partial_y^{-1}(yg_t) = \tfrac{1}{2}(y\, \partial_y^{-1} g_t(y, z, t) - \partial_y^{-2} g_t(y, z, t)), \tag{95}$$

$$h(y, z, t) = \tfrac{1}{2}z\, \partial_z^{-1} g_t(y, z, t), \tag{96}$$

$$h(y, z, t) = \tfrac{1}{2}\partial_z^{-1}(zg_t) = \tfrac{1}{2}(z\, \partial_z^{-1} g_t(y, z, t) - \partial_z^{-2} g_t(y, z, t)). \tag{97}$$

This shows how h can be written in terms of g. For every conservation law corresponding to the densities in Table 3, h could be computed using one of the equations in (94)-(97).

Conservation laws for the KZ equation have been reported in the literature by Sharomet (1989) and Sanders and Wang (1997a). However, substitution of their results into (2) revealed inaccuracies. After bringing the mistake to their attention, Sanders and Wang (1997b) have since corrected one of their conservation laws to match our result.

6.3. The Camassa-Holm Equation in 2 D

The (2+1)-dimensional CH equation,

$$(u_t + \kappa u_x - u_{t2x} + 3uu_x - 2u_x u_{2x} - uu_{3x})_x + u_{2y} = 0, \tag{98}$$

for $u(\mathbf{x}) = u(x, y, t)$ models water waves (Johnson, 2002). It is an extension of the completely integrable 1 D CH equation derived by Camassa and Holm (1993). A study by Gordoa *et al.* (2004) concluded that (98) is not completely integrable.

Obviously, (98) is a conservation law itself,

$$\mathcal{D}_t(u_x - u_{3x}) + \mathcal{D}_x(\kappa u_x + 3uu_x - 2u_x u_{2x} - uu_{3x}) + \mathcal{D}_y(u_y) = 0. \tag{99}$$

It can be written as a system of evolution equations in y. Indeed,

$$u_y = v, \quad v_y = -(\alpha u_t + \kappa u_x - u_{t2x} + 3\beta uu_x - 2u_x u_{2x} - uu_{3x})_x. \tag{100}$$

Note that we introduced auxiliary parameters α and β as coefficients of the u_t and uu_x terms, respectively. The reason for doing so is that the CH equation (98) does not have a scaling symmetry unless we add scales on the parameters α, β and κ. Our code guided us in finding the following conservation laws with functional coefficients,

$$\begin{aligned} &\mathcal{D}_t\Big(fu\Big) + \mathcal{D}_x\Big(\tfrac{1}{\alpha} f(\tfrac{3}{2}\beta u^2 + \kappa u - \tfrac{1}{2}u_x^2 - uu_{2x} - u_{tx}) + (\tfrac{1}{2}f'y^2 - \tfrac{1}{\alpha} fx)(\alpha u_t + \kappa u_x \\ &\quad + 3\beta uu_x - 2u_x u_{2x} - uu_{3x} - u_{t2x})\Big) + \mathcal{D}_y\Big((\tfrac{1}{2}f'y^2 - \tfrac{1}{\alpha}fx)u_y - f'yu\Big) = 0, \end{aligned} \tag{101}$$

$$\begin{aligned} &\mathcal{D}_t\Big(fyu\Big) + \mathcal{D}_x\Big(\tfrac{1}{\alpha} fy(\tfrac{3}{2}\beta u^2 + \kappa u - \tfrac{1}{2}u_x^2 - uu_{2x} - u_{tx}) + y(\tfrac{1}{6}f'y^2 - \tfrac{1}{\alpha}fx)(\alpha u_t + \kappa u_x \\ &\quad + 3\beta uu_x - 2u_x u_{2x} - uu_{3x} - u_{t2x})\Big) + \mathcal{D}_y\Big(y(\tfrac{1}{6}f'y^2 - \tfrac{1}{\alpha}fx)u_y + (\tfrac{1}{\alpha}fx - \tfrac{1}{2}f'y^2)u\Big) = 0, \end{aligned} \tag{102}$$

where $f(t)$ is arbitrary and without constraints on the parameters. Thus, if we set $\alpha = \beta = 1$, we have conservation laws for (98).

6.4. The Gardner Equation in 2 D

The (2+1)-dimensional Gardner equation Konopelchenko and Dubrovsky (1984),

$$u_t = -\tfrac{3}{2}\alpha^2 u^2 u_x + 6\beta uu_x + u_{3x} - 3\alpha u_x \partial_x^{-1} u_y + 3\partial_x^{-1} u_{2y}, \tag{103}$$

for $u(\mathbf{x}) = u(x, y, t)$ is a 2 D generalization of

$$u_t = -\tfrac{3}{2}\alpha u^2 u_x + 6\beta uu_x + u_{3x}, \tag{104}$$

which is an integrable combination of the KdV and mKdV equations due to Gardner. For $\alpha = 0$, (103) reduces to the KP equation (12). For $\beta = 0$, (103) becomes a modified KP equation. Adding a new dependent variable, $v = \partial_x^{-1} u_y$, allows one to remove the integral terms and replace (103) by the system

$$u_y = v_x, \quad v_y = \tfrac{1}{3} u_t - \tfrac{1}{3} u_{3x} - 2\beta u u_x + \alpha u_x v + \tfrac{1}{2}\alpha^2 u^2 u_x. \tag{105}$$

For (103), we found two conservation laws with constant coefficients,

$$\mathcal{D}_t\Big(u\Big) + \mathcal{D}_x\Big(\tfrac{1}{2}\alpha^2 u^3 - 3\beta u^2 + 3\alpha u v - u_{2x}\Big) + \mathcal{D}_y\Big(-(\tfrac{3}{2}\alpha u^2 + 3v)\Big) = 0, \tag{106}$$

$$\mathcal{D}_t\Big(u^2\Big) + \mathcal{D}_x\Big(\tfrac{3}{4}\alpha^2 u^4 - 4\beta u^3 + 3\alpha u^2 v + 3v^2 + u_x^2 - 2u u_{2x}\Big) + \mathcal{D}_y\Big(-u(\alpha u^2 + 6v)\Big) = 0 \tag{107}$$

Using the methodology described for the previous examples in this section, we eventually found three conservation laws involving a variable coefficient $f(t)$,

$$\begin{aligned} &\mathcal{D}_t\Big(fu\Big) + \mathcal{D}_x\Big(f(\tfrac{1}{2}\alpha^2 u^3 - 3\beta u^2 + 3\alpha u v - u_{2x}) + f' y v\Big) \\ &\quad + \mathcal{D}_y\Big(-f(\tfrac{3}{2}\alpha u^2 + 3v) - f' y u\Big) = 0, \end{aligned} \tag{108}$$

$$\begin{aligned} &\mathcal{D}_t\Big(u(fu + \tfrac{2}{3\alpha} y f')\Big) + \mathcal{D}_x\Big(f(\tfrac{3}{4}\alpha^2 u^4 - 4\beta u^3 + 3\alpha u^2 v + 3v^2 + u_x^2 - 2u u_{2x}) \\ &\quad + \tfrac{2}{3\alpha} y f'(\tfrac{1}{2}\alpha^2 u^3 - 3\beta u^2 + 3\alpha u v - u_{2x}) + \tfrac{1}{\alpha}(2x f' + \tfrac{1}{3} y^2 f'')v\Big) \\ &\quad + \mathcal{D}_y\Big(-fu(\alpha u^2 + 6v) - \tfrac{1}{\alpha} y f'(\alpha u^2 + 2v) - \tfrac{1}{\alpha}(\tfrac{1}{3} y^2 f'' + 2x f')u\Big) = 0, \end{aligned} \tag{109}$$

and

$$\begin{aligned} &\mathcal{D}_t\Big((\tfrac{\alpha}{6} y f' + \beta f)u^2 + \tfrac{1}{3}(\tfrac{1}{6} y^2 f'' + x f')u\Big) + \mathcal{D}_x\Big((\tfrac{\alpha}{6} y f' + \beta f)(\tfrac{3}{4}\alpha^2 u^4 - 4\beta u^3 \\ &\quad + 3\alpha u^2 v + 3v^2 + u_x^2 - 2u u_{2x}) + \tfrac{1}{3}(\tfrac{1}{6} y^2 f'' + x f')(\tfrac{1}{2}\alpha^2 u^3 - 3\beta u^2 + 3\alpha u v - u_{2x}) \\ &\quad + \tfrac{1}{3} f' u_x + \tfrac{1}{3} y(\tfrac{1}{18} y^2 f''' + x f'')v\Big) + \mathcal{D}_y\Big(-(\tfrac{\alpha}{6} y f' + \beta f)(\alpha u^2 + 6v)u \\ &\quad - \tfrac{1}{2}(\tfrac{1}{6} y^2 f'' + x f')(\alpha u^2 + 2v) - \tfrac{1}{3} y(\tfrac{1}{18} y^2 f''' + x f'')u\Big) = 0. \end{aligned} \tag{110}$$

Setting $f(t) = 1$ in (108) and (109) yields (106) and (107), respectively.

7. Using the Program ConservationLawsMD.m

Before using ConservationLawsMD.m, all data files provided with the program, as well as additional data files created by the user, must be placed into one directory. Next, open the *Mathematica* notebook ConservationLawsMD.nb which contains instructions for loading the code. Executing the command ConservationLawsMD[] will open a menu, offering the choice of computing conservation laws for a PDE from the menu or from a data file prepared by the user. All PDEs listed in the menu have matching data files. An example of a data file is shown in Figure 1.

The independent space variables must be x, y, and z. The symbol t must be used for time. Dependent variables must be entered as u_i, $i = 1, \ldots, N$, where N is the number of dependent variables. In a (1+1)-dimensional case, the dependent variables (in *Mathematica* syntax) are `u[1][x,t]`, `u[2][x,t]`, etc. In a (3+1)-dimensional cases, `u[1][x,y,z,t]`, `u[2][x,y,z,t]`, etc., where t is always the last argument.

8. Conclusions

We have presented an algorithm and a software package, ConservationLawsMD.m, to compute conservation laws of nonlinear polynomial PDEs in multiple space dimensions.

In contrast to the approach taken by researchers working with *Maple* and *Reduce*, our algorithm uses only tools from calculus, the calculus of variations, linear algebra, and differential geometry. In particular, we do *not* first compute the determining PDEs for the density and the flux components and then attempt to solve these PDEs. Although restricted to polynomial conservation laws, our constructive method leads to short densities (free of divergences and divergence-equivalent terms) and curl-free fluxes.

The software is easy to use, runs fast, and has been tested for a variety of multi-dimensional nonlinear PDEs, demonstrating the versatility of the code. Many of the test cases have been added to the menu of the program. In addition, the program allows the user to test conservation laws either computed with other methods, obtained from the literature, or conjectured after work with the code. The latter is particularly relevant for finding conservation laws involving arbitrary functions as shown in Sections 5 and 6.

Currently, ConservationLawsMD.m has two major limitations: (i) the PDE must either be an evolution equation or correspond to a system of evolution equations, perhaps after an interchange of independent variables or some other transformation; and (ii) the program can only generate local polynomial densities and fluxes. However, the *testing* capabilities of ConservationLawsMD.m are more versatile. The code can be used to test conservation laws involving smooth functions of the independent variables and the densities and fluxes are not restricted to polynomial differential functions.

Future versions of the code will work with any number of independent variables and will cover PDEs that are not of evolution type, e.g., PDEs with mixed derivatives and transcendental nonlinearities.

Acknowledgements

This material is based in part upon work supported by the National Science Foundation (NSF) under Grant No. CCF-0830783. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of NSF.

Mark Hickman (University of Canterbury, Christchurch, New Zealand) and Bernard Deconinck (University of Washington, Seattle) are gratefully acknowledged for valuable discussions. Undergraduate students Jacob Rezac, John-Bosco Tran, and Travis "Alan" Volz

are thanked for their help with this project. We thank the anonymous referees whose constructive comments and suggestions helped us further improve the manuscript.

References

Ablowitz, M.J., Clarkson, P.A., 1991. Solitons, Nonlinear Evolution Equations and Inverse Scattering, Cambridge University Press, Cambridge, U.K.

Ablowitz, M.A., Segur, H., 1981. Solitons and the Inverse Scattering Transform, SIAM Stud. in Appl. Math., vol. 4, SIAM, Philadelphia, Pennsylvania.

Anderson, I.M., 2004a. The Variational Complex. Dept. of Mathematics, Utah State University, Logan, Utah, 318 pages, manuscript available at http://www.math.usu.edu/∼fg_mp/Publications/VB/vb.pdf.

Anderson, I.M., 2004b. The Vessiot package; the software with documentation is available at http://www.math.usu.edu/∼fg_mp/Pages/SymbolicsPage/VessiotDownloads.html.

Anderson, I.M., Cheb-Terrab, E., 2009. DifferentialGeometry package, Maple Online Help, www.maplesoft.com/support/help/Maple/view.aspx?path=DifferentialGeometry.

Baldwin, D., Hereman, W., 2010. A symbolic algorithm for computing recursion operators of nonlinear PDEs, Int. J. Comp. Math. 87, 1094–1119.

Bluman, G.W., Cheviakov, A.F., Anco, S.C., 2010. Applications of Symmetry Methods to Partial Differential Equations, Appl. Math. Sciences, vol. 168, Springer Verlag, New York.

Camassa, R., Holm, D.D., 1993. An integrable shallow water equation with peaked solutions, Phys. Rev. Lett. 71, 1661–1664.

Cheb-Terrab, E., and von Bulow, K., 2004. PDEtools package, Maple Online Help, http://www.maplesoft.com/support/help/Maple/view.aspx?path=PDEtools.

Cheviakov, A.F., 2007. GeM software package for computation of symmetries and conservation laws of differential equations, Comp. Phys. Commun. 76, 48–61.

Cheviakov, A.F., 2010. Computation of fluxes of conservation laws, J. Engr. Math. 66, 153–173.

Deconinck, B., Nivala, M., 2009. Symbolic integration and summation using homotopy operators, Math. Comput. Simul. 80, 825–836.

Drinfel'd, V.G., Sokolov, V.V., 1985. Lie algebras and equations of Korteweg-de Vries type, J. Sov. Math. 30, 1975–2036.

Göktaş, Ü., Hereman, W., 1997, Symbolic computation of conserved densities for systems of nonlinear evolution equations, J. Symbolic Comput. 24, 591–621.

Gordoa, P.G., Pickering, A., Senthilvelan, M., 2004. Evidence for the nonintegrability of a water wave equation in 2+1 dimensions, Zeit. für Naturfor. 59a, 640–644.

Hereman, W., 2006. Symbolic computation of conservation laws of nonlinear partial differential equations in multi-dimensions, Int. J. Quant. Chem. 106, 278–299.

Hereman, W., Adams, P.J., Eklund, H.L., Hickman, M.S., Herbst, B.M., 2009. Direct methods and symbolic software for conservation laws of nonlinear equations. In: Yan, Z. (Ed.), Advances in Nonlinear Waves and Symbolic Computation, Nova Science Publishers, New York, pp. 19–79.

Hereman, W., Colagrosso, M., Sayers, R., Ringler, A., Deconinck, B., Nivala, M., Hickman, M.S., 2005. Continuous and discrete homotopy operators and the computation of conservation laws. In: Wang, D., Zheng, Z. (Eds.), Differential Equations with Symbolic Computation, Birkhäuser, Basel, pp. 249–285.

Hereman, W., Deconinck, B., Poole, L.D., 2007. Continuous and discrete homotopy operators: A theoretical approach made concrete, Math. Comput. Simul. 74, 352–360.

Infeld, E., 1985. Self-focusing nonlinear waves, J. Plasma Phys. 33, 171–182.

Johnson, R.S., 2002. Camassa-Holm, Korteweg-de Vries and related models for water waves, J. Fluid Mech. 455, 63–82.

Kadomtsev, B.B., Petviashvili, V.I., 1970. On the stability of solitary waves in weakly dispersive media, Sov. Phys. Dokl. 15, 539–541.

Konopelchenko, B.G., Dubrovsky, V.G., 1984. Some new integrable nonlinear evolution equations in 2+1 dimensions, Phys. Lett. A 102, 15–17.

Lax, P.D., 1968. Integrals of nonlinear equations of evolution and solitary waves, Commun. Pure Appl. Math. 21, 467–490.

Miura, R.M., Gardner, C.S., Kruskal, M.D., 1968. Korteweg-de Vries equation and generalizations II. Existence of conservation laws and constants of motion, J. Math. Phys., 9, 1204–1209.

Naz, R., 2008. Symmetry solutions and conservation laws for some partial differential equations in field mechanics, Ph.D. dissertation, University of the Witwatersrand, Johannesburg.

Naz, R., Mahomed, F.M., Mason, D.P., 2008. Comparison of different approaches to conservation laws for some partial differential equations in fluid mechanics, Appl. Math. Comput. 205, 212–230.

Newell, A.C., 1983. The history of the soliton, J. Appl. Mech. 50, 1127–1138.

Olver, P.J., 1993. Applications of Lie Groups to Differential Equations, 2nd. ed., Grad. Texts in Math., vol. 107, Springer Verlag, New York.

Poole, L.D., 2009. Symbolic computation of conservation laws of nonlinear partial differential equations using homotopy operators, Ph.D. dissertation, Colorado School of Mines, Golden, Colorado.

Poole, D., Hereman, W., 2009. HomotopyIntegrator.m: A *Mathematica* package for the application of the homotopy method for (i) integration by parts of expressions involving unspecified functions of one variable and (ii) the inversion of a total divergence involving unspecified functions of two or three independent variables; software available at http://inside.mines.edu/∼whereman under scientific software.

Poole, D., Hereman, W., 2009. ConservationLawsMD.m: A *Mathematica* package for the symbolic computation of conservation laws of polynomial systems of nonlinear PDEs in multiple space dimensions, software available at http://inside.mines.edu/∼whereman under scientific software.

Poole, D., Hereman, W., 2010. The homotopy operator method for symbolic integration by parts and inversion of divergences with applications, Appl. Anal. 87, 433–455.

Rosenhaus V., 2002. Infinite symmetries and conservation laws, J. Math. Phys. 43, 6129–6150.

Sanders, J., Wang, J.P., 1997a. Hodge decomposition and conservation laws, Math. Comput. Simul. 44, 483–493.

Sanders, J., Wang, J.P., 1997b. Hodge decomposition and conservation laws; corrected paper, see URL http://www.math.vu.nl/∼jansa/#research.

Sanz-Serna, J.M., 1982. An explicit finite-difference scheme with exact conservation properties, J. Comput. Phys. 47, 199–210.

Sharomet, N.O., 1989. Symmetries, invariant solutions and conservation laws of the nonlinear acoustics equation, Acta Appl. Math. 15, 83–120.

Shivamoggi, B.K., Rollins, D.K., Fanjul, R., 1993. Analytic aspects of the Zakharov-Kuznetsov equation, Phys. Scripta 47, 15–17.

Tikhonov, A.N., Samarskii, A.A., 1963. Equations of Mathematical Physics, Dover Publications, New York.

Vinogradov, A.M., 1989. Symmetries and Conservation Laws of Partial Differential Equations: Basic Notions and Results, Acta Appl. Math. 15, 3–21.

Wolf, T., 2002. A comparison of four approaches to the calculation of conservation laws, Europ. J. Appl. Math. 13, 129–152.

Zakharov, V.E., Kuznetsov, E.A., 1974. Three-dimensional solitons, Sov. Phys. JETP 39, 285–286.

Zakharov, V.E., Shabat, A.B., 1972. Exact theory of two-dimensional self-focusing and one-dimensional self-modulation of waves in nonlinear media, Sov. Phys. JETP 34, 62–69.

```
(* data file d_kd2d.m *)
(* Menu item 2-10 *)

(*** 2D Gardner equation from Konopelchenko and Dubrovsky (1984) ***)

eq[1] = D[u[1][x,y,t],y] - D[u[2][x,y,t],x];

eq[2] = D[u[2][x,y,t],y] -(1/3)*D[u[1][x,y,t],t] + (1/3)*D[u[1][x,y,t],x,3]
 +2*beta*u[1][x,y,t]*D[u[1][x,y,t],x] -alpha*D[u[1][x,y,t],x]*u[2][x,y,t]
 -(1/2)*alpha^2*u[1][x,y,t]^2*D[u[1][x,y,t],x];

diffFunctionListINPUT = {eq[1],eq[2]};
numDependentVariablesINPUT = 2;
independentVariableListINPUT = {x,y};
```

The space variables only; ignore t.

```
nameINPUT = "(2+1)-dimensional Gardner equation";
noteINPUT = "Any additional information can be put here.";

parametersINPUT = {alpha};
```

All parameters without scaling must be placed in this list.

```
weightedParametersINPUT = {beta};
```

Parameters that should have a scaling factor must be placed in this list.

```
userWeightRulesINPUT = {};
```

Optional: the user can choose scales for variables.

```
rankRhoINPUT = Null;
```

Can be changed to a list of values if the user wishes to work with several ranks at once. The program runs automatically when such values are given.

```
explicitIndependentVariablesInDensitiesINPUT = Null;
```

Can be set to $0, 1, 2, \ldots$, specifying the maximum degree $(m+n+p)$ of coefficients $c_i\, t^m x^n y^p$ in the density.

```
formRhoINPUT = {};
```

The user can give a density to be tested. However, this works only for evolution equations in variable t.

```
(* end of data file d_kd2d.m *)
```

Figure 1: Data file for the 2 D Gardner equation in (103).